A Preliminary Cyber-Physical Security Assessment of the Robot Operating System (ROS)

Conference Paper *in* Proceedings of SPIE - The International Society for Optical Engineering · May 2013

DOI: 10.1117/12.2016189

4 authors, including:

Jarrod R. Mcclean
Lawrence Berkeley National Laboratory
49 PUBLICATIONS **1,519** CITATIONS
SEE PROFILE

Charles Farrar
Los Alamos National Laboratory
438 PUBLICATIONS **20,110** CITATIONS
SEE PROFILE

A Preliminary Cyber-Physical Security Assessment of the Robot Operating System (ROS)

Jarrod McClean[a], Christopher Stull[b], Charles Farrar[c], David Mascareñas*[c]
[a]Chemical Physics, Harvard University, 1350 Massachusetts Ave, Cambridge, MA, USA 02138;
[b]Los Alamos National Laboratory, PO Box 1663 MS P915, Los Alamos , NM, USA 87544;
[c]Los Alamos National Laboratory, PO Box 1663 MS T001, Los Alamos , NM, USA 87544

ABSTRACT

Over the course of the last few years, the Robot Operating System (ROS) has become a highly popular software framework for robotics research. ROS has a very active developer community and is widely used for robotics research in both academia and government labs. The prevalence and modularity of ROS cause many people to ask the question: "What prevents ROS from being used in commercial or government applications?" One of the main problems that is preventing this increased use of ROS in these applications is the question of characterizing its security (or lack thereof). In the summer of 2012, a crowd sourced cyber-physical security contest was launched at the cyber security conference DEF CON 20 to begin the process of characterizing the security of ROS. A small-scale, car-like robot was configured as a cyber-physical security "honeypot" running ROS. DEFFCON-20 attendees were invited to find exploits and vulnerabilities in the robot while network traffic was collected. The results of this experiment provided some interesting insights and opened up many security questions pertaining to deployed robotic systems. The Federal Aviation Administration is tasked with opening up the civil airspace to commercial drones by September 2015 and driverless cars are already legal for research purposes in a number of states. Given the integration of these robotic devices into our daily lives, the authors pose the following question: "What security exploits can a motivated person with little-to-no experience in cyber security execute, given the wide availability of free cyber security penetration testing tools such as Metasploit?" This research focuses on applying common, low-cost, low-overhead, cyber-attacks on a robot featuring ROS. This work documents the effectiveness of those attacks.

Keywords: Cyber-physical security, robotic security, honeypot

1. INTRODUCTION

The introduction of cyber-physical systems, such as driverless cars and aerial robotic platforms, into our daily lives will require new tools to ensure their security. Cyber-physical systems feature unique vulnerabilities arising from the intimate coupling of sensors, actuators, mobility, human-to-machine interfaces, and information processing software. New vulnerabilities are emerging that exploit both the cyber *and* physical nature of these devices. For instance, software maliciously injected into a cyber-physical system could be designed in such a way that it becomes active and physically alters its environments when a trigger stimulus is presented to its sensors. Unattended cyber-physical systems are exposed to a wide variety of threats including theft, vandalism, data tampering, contamination, malicious code injection, and sensors spoofing. New threats will also evolve. For instance, an unattended cyber-physical system could become a cyber-physical Trojan horse. Adversarial agents could plant malicious code, sensors, and devices on board the system. When the cyber-physical system returns to its point-of-origin, the security of the point-of-origin could be severely compromised. Furthermore, there are traditional cyber security threats that may only pose a nuisance when they infect a typical computer, but may pose serious safety problems when they infect a cyber-physical system. To date, the cyber-physical security problem is poorly understood. Currently, very few examples of cyber-physical security exploits exist, but those that do exist are quite significant, as will be elaborated upon in the following section. In order to adequately address emerging cyber-physical security challenges, it is necessary to collect data that captures the nature of these threats for further consideration.

The goal of this work is to introduce a new research tool to facilitate cyber-physical security research. This tool is known as the cyber-physical security "honeypot." Like a conventional cyber security honeypot, the cyber-*physical* security honeypot is designed in such a way that it monitors the attempts of the unauthorized use of its cyber-physical resources. The difference is the honeypot also features sensors and actuators to allow for a new class of vulnerabilities and exploits. This initial incarnation of the cyber-physical security honeypot was designed to make use of the popular

Unmanned Systems Technology XV, edited by Robert E. Karlsen, Douglas W. Gage,
Charles M. Shoemaker, Grant R. Gerhart, Proc. of SPIE Vol. 8741, 874110

Robot Operating System (ROS). The hope is that by making use of ROS, its security vulnerabilities can also be discovered and eventually addressed, encouraging wider adoption for commercial and government applications.

In order to quickly collect cyber-physical security vulnerability data at minimal cost, the honeypot was deployed in conjunction with crowd sourcing techniques. It was decided that the cyber-physical security honeypot would be brought to the DEF CON 20 cyber security conference, which typically addresses cyber, as well as physical security threats, including password cracking, wireless network infiltration, lock picking, and social engineering. The cyber-physical security honeypot was presented as a contest challenge at DEF CON 20 where attendees were invited to find and exploit the cyber-physical security vulnerabilities of the honeypot. During the contest, network traffic to the cyber security honeypot was monitored and logged for subsequent analysis purposes.

The goal of this work is to better understand cyber-physical security threats. This knowledge will be essential in the development of strategies to mitigate these threats, and will facilitate widespread adoption of cyber-physical systems.

2. BACKGROUND

Over the course of the last few years, the need for improved cyber-physical security for mobile sensor nodes has been highlighted many times. The U.S. 2009-2034 Unmanned Systems Integrated Roadmap [1] specifically points out that protecting ground-based mobile sensor nodes from tampering is a high priority: "Of particular note among ground systems is the requirement for anti-tampering. In no other environment is an unmanned system more vulnerable to human tampering than when on the ground." The 2011 Unmanned Ground Systems Roadmap identifies technology needs such as a "Render Useless Mechanism." Such a mechanism could be important to destroy data or hardware that could damage the interests of the mobile sensor node owner in the event that it falls under adversarial control. The document also mentions the need for "Layered Escalating Defense Mechanisms," which are described as a "non-lethal, intrusion prevention, and layered, escalating self-defense capability" [2]. Despite these calls for increased cyber-physical security, the authors have found that very little effort has been invested in directly addressing these challenges. There will significant, unexpected cyber-physical security concerns once Unmanned Aerial Systems occupy civil airspace, as will be the case in the next few years (e.g. unauthorized surveillance, property damage, threats to human life). In order to ensure the safety and security of the public when interacting with these systems, it is imperative that we address the emerging cyber-physical security challenges associated with these cost-saving, transformational technologies.

Over the last decade, domestic-task robots have begun to make a debut in homes. The cyber-physical security issues surrounding these devices were studied by Denning [3]. One noteworthy observation Denning makes is that attacks against computing systems tend to lag behind the technology. This implies there is a time window in which security fixes can be made before malicious exploits are developed. The somewhat related problem of identifying cyber-physical security weaknesses in a modern automobile has also been explored [4]. This research showed that it was possible to hack into a modern automobile and sabotage key systems, including the instrument panel cluster, door locks, brakes, and engine control module. More recently, the problem of protecting an autonomous car-like robot from attacks, based on the Precision Immobilization Technique (PIT maneuver), was initially explored by Mascareñas et al. [5], [6]. He found that relatively simple techniques could be used to compromise the physical security of a moving, car-like mobile sensor node. The threat to unmanned vehicles, guided by GPS to spoofing threats, has been explored by Humphries [7]. The findings of this research coupled with the rapid growth of mobile sensor node applications, make it imperative that we begin to address the cyber-physical security challenges that will arise from mobile sensor node use.

3. DESCRIPTION OF THE CYBER+PHYSICAL SECURITY HONEYPOT

In an effort to emulate the cyber-physical challenges associated with deployed mobile systems, the authors repurposed the mobile sensor node employed in [5], [6], to serve as a surrogate for a car-like robotic system. The robot's base is constructed from a commercially available radio controlled truck (Figure 1). On this base, two cameras and a compass are attached to serve as the external input into the system. These inputs are attached to a single board computer running the Linux operating system. The scaled down nature of the cyber-physical security honeypot allows cyber-physical security experimentation to take place in a low-cost manner, with minimal possibility of damage to people or property.

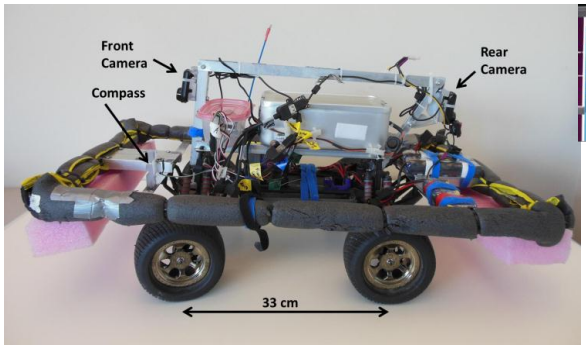

Figure 1. Cyber-physical security honeypot hardware.

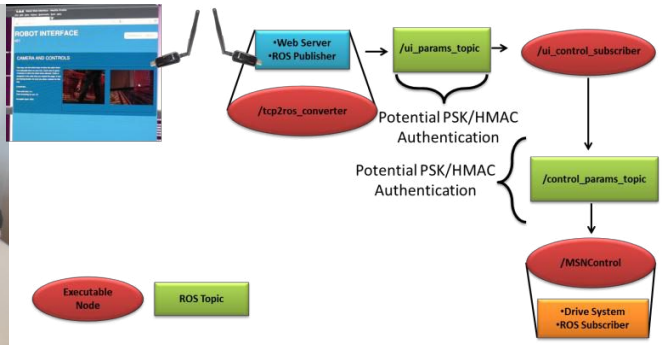

Figure 2. Honeypot software architecture.

The Robot Operating System [8] (ROS) coordinates communication between the hardware and operational software on the robot by means of a message-passing construct. ROS-based robotic systems comprise "nodes" that communicate by publishing "messages" to different "topics," to which other nodes may listen. The system comprises three primary nodes that make the operation of the robot possible. A schematic of the setup is shown in Figure 2, and a detailed description of each of the nodes follows.

In any crowd sourcing experiment, it is critical to maximize participation. As such, user interaction should not require the user to compile or install any additional software. For this reason, the external ROS node serves a webpage, using only JavaScript, to receive user input and translate that input into an equivalent ROS message that could then be sent to any other node in the system. More specifically, the interface is written in standard HTML5 that uses Asynchronous JavaScript and XML (AJAX) to handle user input, and a Python backend is used to identify users and manage a queue system such that only a single user can interact with the system at a time. At this juncture, standard security associated with HTTPS [9] can be used to enhance the security of the user interactions, but keep in mind that this research focused on a *short-term* cyber-physical security honeypot. A benefit of the modularity of the ROS interface, however, is that this external node could be, and was designed to be decoupled from the physical robot in the long term. This node may, for example, be hosted on an enterprise grade web-server that serves as a gateway from an external network to the robot, which receives ROS messages from the gateway wirelessly. This portable and modular setup allows the crowd sourcing experiment to extend globally, potentially allowing millions of participants. Future work will look at building such a long-term cyber-physical security honeypot.

The translation node, which receives messages from external nodes, is responsible for processing of raw user and external input into intelligible internal commands. This node serves an essential role in both security and human interaction. From a security point of view, it is essential that the messages translated by this node be verified, such that no message is passed on to the physical system that can cause unintended harm to the robot or its surroundings. Even an authorized user of a system may accidently use a robot to perform beyond its capabilities, and this node must interpret the user's requests within the context of the robot's design. In this respect, it acts as input sanitation, which is a vital part of many traditional database and software systems. However, the key difference between this node and traditional input sanitation is that this node may also need to sanitize raw hardware input (such as camera data), which may be suspect in a hostile environment.

The most internal node of the system, the drive node, accepts ROS messages from the translation node and triggers the drive system. This form of modularity allows external kill switches and safety measures to communicate to only one node, whose termination ensures safe stopping conditions and does not interfere with other internal functions of the robot.

4. KNOWN VULNERABILITIES

While a primary goal of the project was to expose unknown vulnerabilities in cyber-physical systems constructed with the ROS framework, the authors were aware of several vulnerabilities and purposefully left those vulnerabilities in place to facilitate the experiment and simulate more realistic field conditions. Among these vulnerabilities are plain-text communications, unprotected TCP ports, and unencrypted data storage.

By default, both the external web interface and ROS node-to-node communications are done in plain text. This has certain benefits for ease of use, debugging, and performance, as additional cryptographic steps are not required. In the case of the external web interface, transitioning to HTTPS security would be relatively seamless with modern web frameworks; however, assessing the security of the HTTPS protocol was not the intention of this experiment, and only represented an additional barrier for the user to access the robot. For ROS node-to-node communication, it is evident that the plain-text nature of the messages allows an unauthorized user to easily interpret the form of the message, and spoof fake messages. One possible solution to this is discussed later in this work.

The internal communication structure of ROS is built around TCP ports, allowing for a modular robot to be distributed across the world if desired. However, the downside of this modularity is the exposure of TCP ports that offer little authentication, at the moment. In this experiment, because all nodes were internal to the physical robot, the authors could have dropped external packets arriving at the ROS ports. However, to better study the risks associated with a truly distributed system, these ports were left open. This offers a more true simulation of a complex system where it is difficult to verify each individual path of communication.

It is clear that in many applications, it is undesirable for hardware failure to result in the average passerby being able to recover sensitive or proprietary data off of a storage device. As such, encryption of a hard drive or any equivalent data storage is a common sense procedure that the authors expect will be followed in all deployed cyber-physical systems. However, for research purposes, it is sometimes desirable to be able to recover data in extreme circumstances such as hardware failure, which would be hindered by maintaining encrypted data. As a result, the hard drive and all data storage on the device are left unencrypted.

5. AUTHENTICATION SOLUTION

A key security point touched upon earlier in this work is the plain-text nature of ROS messages. Not only can a third party easily decipher the messages, but they can also be easily spoofed, as was demonstrated by a participant familiar with ROS. This has serious implications when the result of an erroneous message can be to drive a cyber-physical system into a building or wall. As such, it is critical that received messages can be authenticated and that the identity of the sender can be verified.

As such, a simple fix is to utilize standard hash-based message authentication code (HMAC) methods to add a hash, based on a pre-shared key between each of the nodes. If established HMAC protocols [10] are followed with a sufficiently strong cryptographic hash, each message can be quickly authenticated and the identity of the senders (each node) can be verified. These requirements are not met with simple encryption algorithms such as DES, as bits within the encrypted data could be maliciously altered, causing unanticipated values on decryption. A combination of the two could, of course, be used. However, if one is limited in computational power, as is frequently the case in deployed, embedded systems, it is clear that in cyber-physical systems, priority must go to message authentication rather than message encryption, as invalid messages could cause physical damage.

Moreover, for computationally intensive tasks, an unencrypted HMAC procedure can make excellent use of modern multicore processor architectures, which typically consume fewer watts per FLOP (floating point operation) than a single core processor run at a faster clock speed. For example, if a complex computation is required on a large data set, two threads can be used simultaneously to begin verifying and computing on the data. If at any point, the authentication fails, the computation thread is terminated, and if the computation thread finishes first, it waits until authentication is completed to perform a physical task or change state values.

6. CROWD SOURCED CONTEST/EXPERIMENT AT DEF CON 20

The cyber-physical security honeypot experiment was set up at the DEF CON 20 cyber security conference in the contest area. The DEF CON contest designation was “Dr. Strange Bot.” This experiment was a late arrival to the conference and does not appear on the DEF CON official program. Figure 3 shows the contest area, which is composed of a cage for the cyber-physical security honeypot, made from plastic orange construction fence in order to ensure the safety of DEF CON 20 attendees.

Figure 3. DEF CON 20 cyber-physical security honeypot contest/experiment setup.

Figure 4. Cyber-physical security honeypot located inside its cage.

Figure 4 shows the cyber-physical security honeypot inside its cage during the contest. The first iteration of this contest/experiment presented DEF CON 20 attendees with the following scenario. It was first assumed that it is the near future when robotic devices are more fully integrated into our daily lives. For instance, robots may be used commercially at hotels as bellhops who would pick up luggage and bring it to a customer's hotel room. Commercial robots would have a legitimate user interface that would be made available to valid users. DEF CON 20 attendees were considered valid users for the purposes of this experiment. As such, they were given the password required to log into the WPA2-protected wireless network used to communicate with the honeypot. Attendees could then access a web-based user interface that could be used to drive the honeypot as well as see the data from its cameras. Figure 5 shows the web-based user interface under normal operating conditions during the contest. DEF CON 20 attendees were then invited to find vulnerabilities and unique exploits to take control of the honeypot, when given access given access to the legitimate user interface. For this initial experiment, the human subjects research review board that oversaw this experiment asked that we ensure all participants remain anonymous. As a result we could not provide any prizes or recognition for creative exploits at this time.

7. SUMMARY OF EXPLOITS AND MALFUNCTIONS

During the course of the DEF CON 20 conference, a number of cyber-physical security exploits were experienced by the honeypot. Two solid state drives used to control the honeypot failed during the course of the experiment. Both of these drives are no longer able to boot and the data on one of the drives is currently not recoverable. At one point during the experiment, a participant with expertise in ROS was able to inject false ROS messages into the robot to actuate the throttle. In order to facilitate this exploit, the person was given details on the ROS message fields. Using this information the person was able to generate false messages in a matter of minutes. The research team decided to give details on the ROS messages to this participant because the experiment was announced on short notice and participants did not have sufficient time to prepare. However, the research team theorizes that someone with a modest background in cyber security could potentially get this information with ease as ROS messages are transmitted in plain text. A subsequent section will explore this hypothesis more thoroughly.

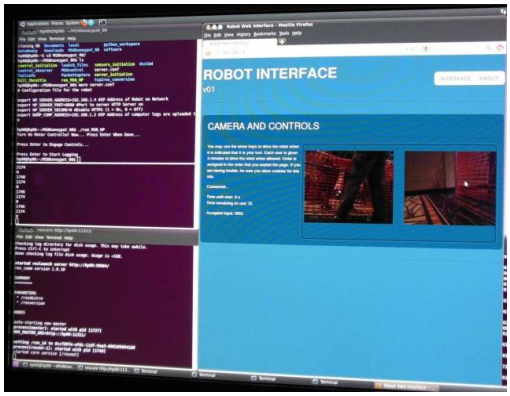

Figure 5 Screenshot from cyber-physical security honeypot user interface during normal operation.

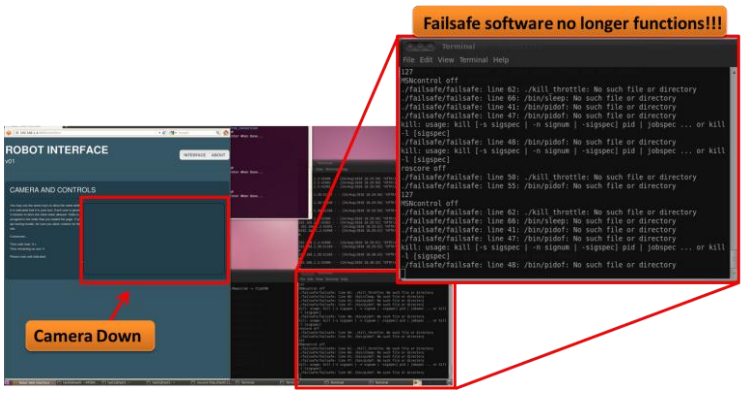

Figure 6 Screenshot from honeypot immediately after the software failsafe went down.

Another notable exploit occurred when the software failsafe on board the cyber-physical security honeypot became inoperable. Concurrently, the front and rear cameras on board the honeypot were no longer able to send data. A screen shot from the honeypot immediately after this exploit took place is shown in Figure 6. At this time it is not clear whether the camera malfunction and the failure of the software failsafe are related. There was a period of about one hour before and after the failure of the software failsafe that the cameras would sporadically go down. Neither of these failures was observed during the development and testing phase of the cyber-physical security honeypot. This experiment illustrates that the complex nature of cyber-physical systems makes it very difficult to discern whether or not these failures are malfunctions inherent to the cyber-physical security honeypot or the result of an adversarial action. The difficulty associated with classifying these events as either malfunctions or adversarial actions illustrates that new tools and techniques may be needed to conduct effective cyber-physical security forensics. The last exploit/malfunction worth mentioning is that shortly after returning to Los Alamos, the base station computer's hard drive failed. The hard drive can no longer boot the computer, and data from the hard drive has resisted all attempts at being recovered.

8. ROS MESSAGE INTERCEPTION USING WIRESHARK

As discussed previously, a participant familiar with ROS was able to spoof ROS messages, thereby operating our ROS-based robotic system *without* using the web-based user interface provided. While spoofing the ROS messages appeared to require a relatively trivial level of effort, a prerequisite of this effort was knowledge of the ROS message headers used to control the system. Further discussion led the authors to hypothesize that an individual could gain access to the required ROS message headers by simply "sniffing" the TCP packets of the robotic system's local host.

Therefore, as a postlude to the cyber-physical security honeypot, the authors posed the following question: "Can an individual with limited experience in the area of cyber security successfully exploit a simulated ROS-based robotic system?" As a preliminary step toward answering this question, the authors first sought to validate their hypothesis pertaining to whether knowledge about ROS message headers could be extracted from TCP packets.

Again, assuming limited experience in the area of cyber security, the authors turned to Wireshark [11], a freely-available network protocol analyzer, to sniff and decipher the TCP packets generated by a simulated ROS-based robotic system from the online ROS tutorials [12]. Figure 7 offers a screen capture from this preliminary effort, clearly showing the ROS message headers "float32 linear" and "float32 angular" as well as the message MD5 checksum. This result lends credence to the validity of the authors' hypothesis. Continuations of this work are underway to further validate this hypothesis on a deployed robotic system and to develop basic packet interception and injection software that, together with the knowledge gained by Wireshark, could be employed to inject spoofed packets into a ROS message stream (a so-called "man-in-the-middle" attack).

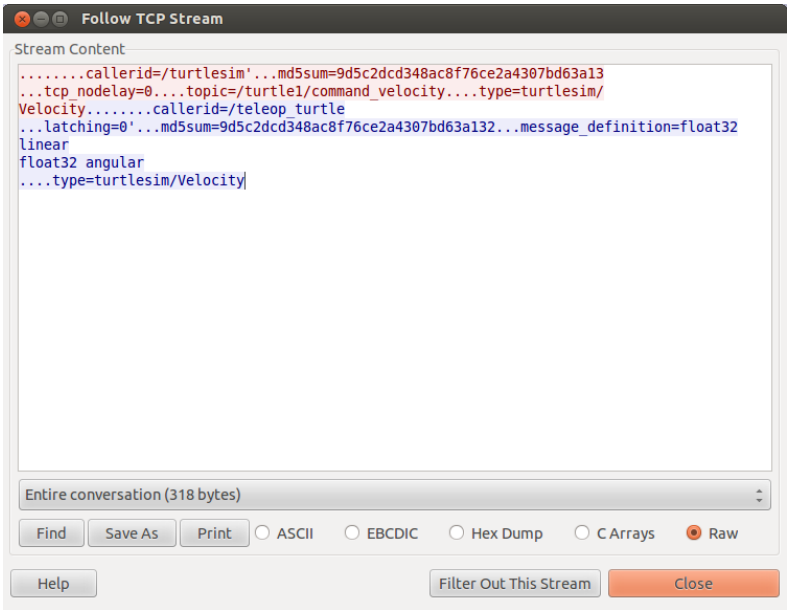

Figure 7 Data collected from Wireshark showing the plain text data transmitted by a ROS message.

9. LESSONS LEARNED

One of the most interesting results of the honeypot experiment is that it can be very difficult to discern between a cyber-physical security exploit and hardware and/or software bugs. The obvious implication of this finding is that the detection and attribution of malicious attacks could be more difficult than expected. Furthermore, this difficulty may be exploited as a means by which exploits can be disguised as simple bugs, while in actuality, the system is under attack.

The authors also noted a number of practical lessons from this study. Among the most practical of these is that the cyber security culture is very "game-oriented." That is, the more the challenge was presented as a game, the more attendees seemed interested in it. This realization is driven home in [13], where it is stated that the "Metasploit project was originally started as a network security game by four core developers."

Another practical lesson is that despite DEF CON being primarily a cyber security conference, few of the participants carried laptops, often opting to carry smart phones instead. Fewer still were willing to connect their laptops to our network: a reasonable behavior given the hostile nature of DEF CON 20 networks. Thus, future honeypot-like experiments will insure that: (1) participant interfaces are compatible with smart phones; and (2) computing terminals, equipped with common cyber security tools (*e.g.* running Backtrack Linux), will be provided at the booth.

During the course of this work, it was realized that in order to secure cyber-physical systems, a new type of security professional will be required. The new class of security professional will need a background in both cyber *and* physical security, as well as robotics. Of all the exploits recorded during this experiment, the most interesting one was executed by a participant who had such a background. In order to get a wider variety of exploits, it may be helpful to form small red teams consisting of people with both robotic as well as cyber security backgrounds. Alternatively, if the honeypot is deployed at another conference, it may be helpful to give short tutorials on both robotics, as well as cyber security before the beginning of the contest. The hope is that by giving participants more background knowledge they may be able to come up with more creative exploits in a shorter period of time.

Lastly, we noted that our test platform should be setup to better protect the base station computer from attacks. Oftentimes throughout the experiment, the authors noticed odd behaviors exhibited by the base station computer (*e.g.* muting and un-muting of the speakers without issuing the mute command). That said, it is still questionable whether this was an exploit or a bug. The team had never observed this behavior outside of DEF CON. Regardless, the next experiment will adopt more secure networking strategies for system-administration-focused computers. Possible strategies include the use of a virtual honey net.

10. CONCLUSIONS

This initial crowd source deployment of the cyber-physical security honeypot at the DEF CON 20 conference provided valuable information to guide future work in the emerging field of cyber-physical security. The results from this contest/experiment will be used to guide the development of a second generation experiment. For example, the next generation experiment will be designed to facilitate physical interaction between the participants and the cyber-physical security honeypot in a manner that draws attention from the cyber security community. Future work will also include the design of long-term, research honeypot experiments/contests.

11. ACKNOWLEDGEMENT

The authors would like to acknowledge the Los Alamos National Laboratory, Laboratory Directed Research and Development program for funding this work. Jarrod McClean was funded by the Department of Energy Computational Science Graduate Fellowship. We would also like to thank the Pyr0, the DEF CON 20 contest organizer for accommodating this contest/experiment on very short notice. We would also like to acknowledge the help of the Los Alamos National Laboratory Human Subjects Research Review Board for reviewing this research and providing valuable suggestions. LANL HSRRB project identifier LANL 12-12 X. LA-UR-13-22529.

REFERENCES

[1] U.S. Department of Defense. FY2009-2034, “Unmanned Systems Integrated Roadmap” (2009).

[2] Robotic Systems Joint Program Office, “Unmanned Ground Systems Roadmap” (2011).

[3] Denning, T., Matuszek, C., Koscher, K., Smith, J., Kohno, T., “A spotlight on security and privacy risks with future household robots: Attacks and lessons.” In Proc. of the 11th International Conference on Ubiquitous Computing (Ubicomp '09); Orlando, Florida (2009).

[4] Kosher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., et al., “Experimental Security Analysis of a Modern Automobile.” In The IEEE Symposium on Security and Privacy; May 16-19, Oakland, CA (2010).

[5] Mascarenas, D., Stull, C., Farrar, C., “Escape and Evade Control Policies for ensuring the physical security of nonholonomic, ground-based, unattended mobile sensor nodes.” In SPIE Defense Security and Sensing; April, Orlando, Florida (2011).

[6] Mascarenas, D., Stull, C., Farrar, C., “Towards the development of tamper-resistant, ground-based mobile sensor nodes,". In SPIE Security and Defense ; Prague, Czech Republic (2011).

[7] Humphreys, TE., Ledvina, BM., Psiaki, M.,L., O'Hanlon, W.B., Kinter, P.M., “Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer.” In Proceedings of ION GNSS, The Institute of Navigation; Savanna, Geogia (2008).

[8] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., et al. “ROS: an open-source Robot Operating System.” In Proc. Open-Source Software workshop of the International Conference on Robotics and Automation ; (2009).

[9] The Internet Society, “HTTP over TLS.” Network Working Group; May, Report No.: RFC 2818 (2000).

[10] The Internet Society, “HMAC: Keyed-Hasing for Message Authentication.” , Network Working Group; February 1997. Report No.: RFC2104.

[11] Wireshark. http://www.wireshark.org/..

[12] Willow Garage, “ROS Tutorials.” http://www.ros.org/wiki/ROS/Tutorials. (2012).

[13] Maynor D., “Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research,” Burlington: Syngress Publishing, Inc.; (2007).